Software Engineering Methodology

Chapter 10.0
Software Maintenance

Table of Contents

Chapter **Page**

Chapter: **10.0**

Software Maintenance

Description: This chapter describes an iterative process for conducting software maintenance activities. The process prescribes a minimal set of criteria that are necessary for project management and quality assurance processes; control; and management of the planning, execution, and documentation of software maintenance activities. The use of automated tools to facilitate requirements definition, design, coding, and system documentation is encouraged. The selection and implementation of tools varies among sites and organizations.

The basic maintenance process model includes input, process, output, and control for software maintenance. It is based on the same software engineering principles and preferred practices that lower risk and improve quality during the planning and development stages of the lifecycle. The process model supports the concept that planned software changes should be grouped and packaged into scheduled releases that can be managed as projects. This proven approach allows the maintenance team to better plan, optimize use of resources, take advantage of economies of scale, and better control outcome in terms of both schedule and product quality.

Each organization performing software maintenance activities should have a local documented procedure for handling emergency changes that cannot be implemented as part of a scheduled release. Generally, these changes include fixes to correct defects and updates to meet unscheduled business or legal requirements. Emergency changes should be integrated into the next release for full regression testing and documentation updates.

Stages: The activities to be performed during software maintenance are grouped into logically related segments of work called "stages." These stages are similar to those referenced in the planning and development stages of the software lifecycle. The stages are presented in the sections listed below.

10.1 Problem/Modification Identification Stage
10.2 Analysis Stage
10.3 Design Stage
10.4 Programming Stage
10.5 System Test Stage
10.6 Acceptance Stage
10.7 Delivery Stage

A matrix depicting the maintenance process model stages is provided in *Exhibit 10.0-1, Process Model for Software Maintenance.*

Note: The maintenance process model does not presume the use of any particular software development methodology (e.g., waterfall or spiral). This process is valid regardless of size, complexity, criticality, application of the software product, or usage of the software in the system to be maintained.

The software maintenance stages can be tailored (i.e., logically combining stages or outputs) as appropriate. Stages may be combined to more effectively manage the project. Decisions to combine stages are agreed to by the designated approvers during the Analysis Stage. Factors that can influence the number of stages include level of effort, complexity, visibility, and business impact. Guidance to assist with decisions to combine stages is presented in *Exhibit 10.0-2, Tailoring For Size*.

Project Management: To the extent possible, all software maintenance activity should be managed as a project to gain the benefits inherent in project management and to enable tracking of activities and costs. The extent of project management activity will vary, and should be tailored according to the size, complexity, and impact of the change or enhancement.

Review Processes: In each stage, one or more structured walkthroughs are conducted to validate work products. *Appendix C, Conducting Structured Walkthroughs*, provides a procedure and sample forms that can be used for structured walkthroughs.

In software maintenance, and especially for major modifications, one or more In-Stage Assessments are conducted as part of the quality assurance activities for each stage. This process is documented in *Appendix D, In-Stage Assessment (ISA) Process Guide*.

A Stage Exit is conducted at the end of each stage of software maintenance. This process, which includes definition of participant roles and the review and approval process, is documented in *Appendix E, Stage Exit Process Guide*.

Metrics: Metrics/measures and associated factors for each stage should be collected and reviewed at appropriate intervals. *Exhibit 10.0-3, Process Model Metrics for Software Maintenance*, provides metrics for each stage of software maintenance. Metrics/measures captured for maintenance should enhance the implementation and management of this process.

Conventions: The following convention is used in each exhibit depicting a software maintenance stage.

```
                    Control
                          T
                          *
                          ▾
                     +))))))))))))),
      Input  S)))) ▸ *Process Name * S))) ▸ Output
                     .)))))))))))))-
                          T
                          *

                Associated Process
```

An "associated process" is one that is executed in support of software maintenance, but is itself not defined in this chapter (e.g., Stage Exit).

Exhibit 10.0-1. Process Model for Software Maintenance

	Problem Identification Stage	Analysis Stage	Design Stage	Programming Stage	System Test Stage	Acceptance Stage	Delivery Stage
Input	Modification Request (MR)	Project/system document Project file information Validated MR	Project/system document Source code Database(s) Analysis stage output	Source code Product/system document Results of design stage	Updated software Documentation Test Readiness Review Report Updated system	Test Readiness Review Report Integrated system Acceptance test: Plans Cases Procedures	Tested/accepted system
Process	Assign change number Classify Accept or reject change Preliminary effort estimate	Feasibility analysis Detailed analysis Redocument, if needed	Revise: Requirements System doc. Module doc. Project plan Create test cases	Code Unit test Test Readiness Review	Functional test Interface testing Regression testing Test Readiness Review	Acceptance test Interoperability test Functional Configuration Audit (FCA)	Physical Configuration Audit (PCA) Install Training
Output	Validated MR Process determinations	Feasibility Report Detailed Analysis Report Updated: Requirements Modification list Test strategy Project plan	Revised: Modification list Detailed analysis Updated: Design baseline Test plans Project plan	Updated: Software Design documents Test documents User documents Training materials Project plan Test readiness review report	Tested system Test reports Updated project plan	New system baseline Acceptance Test Report FCA Report Updated project plan	PCA Report Version Description Document (VDD)
Review Assurance Approve	Peer review(s)	Structured Walkthrough(s) In-Stage Assessment(s) Stage Exit	Structured Walkthrough(s) In-Stage Assessment(s) Stage Exit	Structured Walkthrough(s) In-Stage Assessment(s) Stage Exit	Structured Walkthrough(s) In-Stage Assessment(s) Stage Exit	Structured Walkthrough(s) In-Stage Assessment(s) Stage Exit	Structured Walkthrough Stage Exit
Metrics	*See Exhibit 10.0-3, Process Model Metrics for Software Maintenance*						

Exhibit 10.0-2. Tailoring For Size

```
                                  SOFTWARE MAINTENANCE STAGES
 SIZE ¹
                       ▼         ▼         ▼         ▼         ▼         ▼         ▼

             *         *         *         *         *         *         *         *
LARGE        /)))))))))3)))))))))3)))))))))3)))))))))3)))))))))3)))))))))3)))))))))1
             *Prob. ID *Analysis *Design   *Program. *Sys.Tst. *Accept.  *Delivery *


                       ▼              ▼                 ▼              ▼

             *         *              *                 *              *
MEDIUM       /)))))))))3))))))))))))))3)))))))))))))))))3))))))))))))))1
             *Prob. ID *Anal./Design  *Program./Test    *Accep./Deliv. *

                                    ▼                         ▼

             *                      *                         *
SMALL        /)))))))))0))))))))))))3))))))))))))))0))))))))))1
             *Prob. ID *Anal./Desi. *Program/Test  *Acc./Del. *
             R         R            R              R          R


           LESS   )))))))))))))))))))))))))))))))))))))))))))))))))Q          ▸ MORE
                       DEGREE OF PROJECT MANAGEMENT REQUIRED
```

▼ = Stage Exit occurs at this point.

[1] Size is used as a guide to help determine the appropriate degree of project management, and which stages may be combined for a given effort. Within this context, size is a combination of level of effort required (all activities) and complexity of the modification. Attributes of complexity include technology, team skills, interfaces, and level of understanding of requirements.
Other factors that can influence tailoring include risk, visibility, and business impact.

Exhibit 10.0-3. Process Model Metrics for Software Maintenance

	Problem Identification Stage	Analysis Stage	Design Stage	Programming Stage	System Test Stage	Acceptance Stage	Delivery Stage
Factors	Correctness Maintainability	Flexibility Traceability Usability Reusability Maintainability Comprehensibility	Flexibility Traceability Reusability Testability Maintainability Comprehensibility Reliability	Flexibility Traceability Maintainability Comprehensibility Reliability	Flexibility Traceability Verifiability Testability Interoperability Comprehensibility Reliability	Flexibility Traceability Interoperability Testability Comprehensibility Reliability	Completeness Reliability
Metrics	No. of omissions on Modification Request (MR) No. of MR submittals No. of duplicate MRs Time expended for problem validation	Requirement changes Documentation error rates Effort per function area (e.g., SQA) Elapsed time (schedule) Error rates, by priority and type	S/W complexity Design changes Effort per function area Elapsed time Test plans and procedure changes Error rates, by priority and type Number of lines of code, added, deleted, modified, tested	Volume/ functionality (function points or lines of code) Error rates, by priority and type	Error rates, by priority and type Generated Corrected	Error rates, by priority and type Generated Corrected	Documentation changes (i.e. version description documents, training manuals, operation guidelines)

Note: The above level of metrics is a goal. Each organization responsible for software maintenance activities should establish an individual plan to achieve this level over time.

Bibliography: The following materials were used in the preparation of the Software Maintenance chapter.

1. DeMarco, Tom, *Controlling Software Projects* , New York, 1989.

2. Frame, Davidson J., *Managing Projects in Organizations* , San Francisco, 1987.

3. Frame, Davidson J., *The New Project Management* , San Francisco, 1994.

4. Page-Jones, Meilir, *Practical Project Management* , New York, 1985.

5. The Institute of Electrical and Electronics Engineers, Inc., *IEEE Standard for Software Maintenance*, IEEE Std 1219-1992, New York, 1993.

6. U.S. Department of Commerce/National Bureau of Standards, *Guideline on Software Maintenance*, Springfield, Virginia, 1984.